

Linguagens Formais e Autômatos

Expressões Regulares (ERs)

Andrei Rimsa Álvares
andrei@cefetmg.br



Expressões Regulares

- Até agora foram vistas duas formas de se especificar uma linguagem regular
 - Usando notação de conjuntos
 - **Aplicação:** desenvolvimento da teoria
 - Desenhando um diagrama de estados em forma de grafo
 - **Aplicação:** processo de concepção de um reconhecedor
- Uma nova forma de especificar linguagem regular é através de **expressões regulares**, mediante uma expressão que a denota
 - **Aplicação:** na teoria para manipulações formais e na prática para referência compacta a conjunto de palavras (editores de texto e comandos de sistema operacional)



Expressões Regulares

- Dado um AF M , seria possível obter uma formulação *linear* para $L(M)$? Ou seja, seria possível obter uma expressão r que denotasse $L(M)$?

Sim, através de uma expressão regular!

- Dada uma expressão regular, é possível criar um AF que reconhece a linguagem denotada por ela
 - Logo, a família das linguagens denotadas por expressões regulares é exatamente a família das linguagens regulares



Expressões Regulares

- Uma **expressão regular** (ER) sobre um alfabeto Σ é definida recursivamente como
 - a) \emptyset , λ e a para qualquer $a \in \Sigma$ são expressões regulares; tais ERs denotam, respectivamente, os conjuntos \emptyset , $\{\lambda\}$ e $\{a\}$
 - b) Se r e s são expressões regulares, então são expressões regulares: $(r+s)$, (rs) e r^* ; tais ERs denotam, respectivamente $L(r) \cup L(s)$, $L(r)L(s)$ e $L(r)^*$

Um conjunto que pode ser denotado por uma ER é usualmente denominado **conjunto regular**.



Exemplo de Expressões Regulares

- Exemplos de ERs sobre o alfabeto $\Sigma = \{0, 1\}$ e os conjuntos regulares denotados por elas

ER	Linguagem denotada
\emptyset	\emptyset
λ	$\{\lambda\}$
(01)	$\{0\}\{1\} = \{01\}$
$(0+1)$	$\{0\} \cup \{1\} = \{0,1\}$
$((0+1)(01))$	$\{0,1\}\{01\} = \{001,101\}$
0^*	$\{0\}^* = \{0^n \mid n \geq 0\}$
$(0+1)^*$	$\{0,1\}^* = \Sigma^*$
$((((0+1)^*1)(0+1))$	$\{0,1\}^*\{1\}\{0,1\}$



Omissão de Parênteses

- Uma ER com muitos parênteses torna difícil sua escrita e compreensão; algumas regras para omissão de parênteses
 - a) Como a união é associativa, pode-se escrever $(r_1 + r_2 + \dots + r_n)$, omitindo-se os parênteses internos
 - Ex.: $((0 + (11)) + 1)$ ou $(0 + ((11) + 1)) \rightarrow (0 + (11) + 1)$
 - b) Idem para concatenação, já que também é associativa
 - Ex.: $((01)((00)1)) \rightarrow (01001)$
 - c) Se a expressão tem parênteses externos, esses podem ser omitidos
 - Ex.: $a) \rightarrow 0 + (11) + 1$
 $b) \rightarrow 01001$
 - d) Precedência: Fecho de Kleene > Concatenação > União
 - Ex. $(0 + (10^*)) \rightarrow 0 + 10^*$



Obtenção de ER que Denote L

- Seja L o conjunto das palavras sobre $\{a, b\}$ com pelo menos um b . Como obter L ?



Como obter uma expressão regular que denote uma linguagem L ?



Obtenção de ER que Denote L

- Seja L o conjunto das palavras sobre $\{a, b\}$ com pelo menos um b . Como obter L ?

Primeira abordagem: tentar visualizar a linguagem como um todo, utilizando **não determinismo** quando possível

- Ex.: uma palavra de L tem um b que é precedido de zero ou mais símbolos (inclusive de b 's: não determinismo) e seguido por zero ou mais símbolos

$$(a + b)^* b (a + b)^*$$

Um mesmo conjunto pode ser denotado por várias ERs?



Obtenção de ER que Denote L

- Seja L o conjunto das palavras sobre $\{a, b\}$ com pelo menos um b . Como obter L ?

Segunda abordagem: visualizar como as palavras podem ser construídas deterministicamente, da esquerda para direita ou da direita para esquerda

- Ex.: da esquerda para direita \rightarrow pertence a L quando se encontra um b

$$a^*b(a + b)^*$$

- Ex.: da direita para esquerda \rightarrow pertence a L quando se encontra um b

$$(a + b)^*ba^*$$



Equivalência entre ERs

- Conforme visto, um conjunto pode ser denotado por várias expressões regulares
- A notação $r = s$ é comum para duas expressões regulares r e s que denotam uma mesma linguagem

$$L(r) = L(s)$$

- Às vezes pode ser interessante obter uma ER "mais simples", no sentido de visualizar com mais facilidade a linguagem por ela denotada
 - As propriedades de **união**, **concatenação** e **fecho de Kleene** permitem obter ERs equivalentes, eventualmente mais simples



Equivalência entre ERs

- Algumas equivalências entre ERs

(1)	$r + s = s + r$	(11)	$r^{**} = r^*$
(2)	$r + \emptyset = r$	(12)	$r^* = (rr)^*(\lambda + r)$
(3)	$r + r = r$	(13)	$\emptyset^* = \lambda$
(4)	$r\lambda = \lambda r = r$	(14)	$\lambda^* = \lambda$
(5)	$r\emptyset = \emptyset r = \emptyset$	(15)	$r^*r^* = r^*$
(6)	$(r + s)t = rt + st$	(16)	$rr^* = r^*r$
(7)	$r(s + t) = rs + rt$	(17)	$(r^* + s)^* = (r + s)^*$
(8)	$(r + s)^* = (r^*s)^*r^*$	(18)	$(r^*s^*)^* = (r + s)^*$
(9)	$(r + s)^* = r^*(sr^*)^*$	(19)	$r^*(r + s)^* = (r + s)^*$
(10)	$(rs)^* = \lambda + r(sr)^*s$	(20)	$(r + s)^*r^* = (r + s)^*$



Equivalência entre ERs

- Qualquer equivalência que não envolva fecho de Kleene pode ser derivada a partir das propriedades 1 a 7 mais as propriedades associativas de união e concatenação
- Quando envolve fecho de Kleene não há um conjunto finito de equivalências das quais se pode derivar qualquer outra
- Existem várias equivalências que são redundantes (podem ser obtidas através de outras)

$$\begin{aligned}\emptyset^* &= (r\emptyset)^* && , \text{ por 5} \\ &= \lambda + r(\emptyset r)^* \emptyset && , \text{ por 10} \\ &= \lambda + \emptyset && , \text{ por 5} \\ &= \lambda && , \text{ por 2}\end{aligned}$$



Simplificação de ERs

- Exemplo de uma série de simplificações de uma ER utilizando as equivalências da tabela, onde as subexpressões simplificadas em cada passo estão sublinhadas

$$\begin{aligned} & \underline{(00^* + 10^*)}0^*(1^* + 0)^* \\ &= (0 + 1)\underline{0^*0^*}(1^* + 0)^* \quad , \text{ por 6} \\ &= (0 + 1)0^*\underline{(1^* + 0)^*} \quad , \text{ por 15} \\ &= (0 + 1)0^*\underline{(1 + 0)^*} \quad , \text{ por 17} \\ &= (0 + 1)\underline{0^*(0 + 1)^*} \quad , \text{ por 1} \\ &= (0 + 1)(0 + 1)^* \quad , \text{ por 19} \end{aligned}$$



Simplificação de ERs

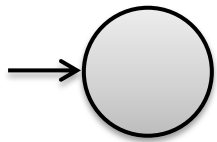
- Novas equivalências podem ser obtidas considerando as propriedades de
 - **justaposição:** concatenação
 - **+**: união
 - *****: fecho de Kleene
- Por exemplo
 - $(r + rr + rrr + rrrr)^* = r^*$
 - $\underline{((0(0 + 1)1 + 11)0^*(00 + 11))^*(0 + 1)^*} = (0 + 1)^*$
Como é da forma $\underline{r}(0+1)^*$, onde r denota um conjunto de palavras sobre o alfabeto que contém λ , então não precisa analisar r
 - $r^*(r + s^*) = r^*s^*$



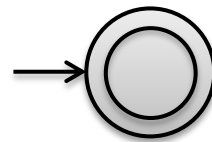
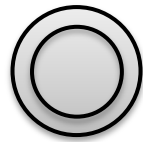
Linguagens Regulares

- Toda **expressão regular** denota uma **linguagem regular**

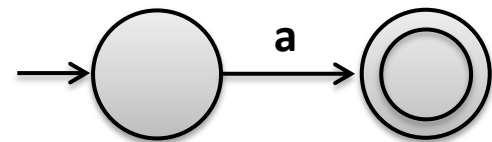
a) É possível construir AFs para os elementos primitivos dos conjuntos regulares: \emptyset , λ e para cada $a \in \Sigma$



AF para \emptyset



AF para $\{\lambda\}$



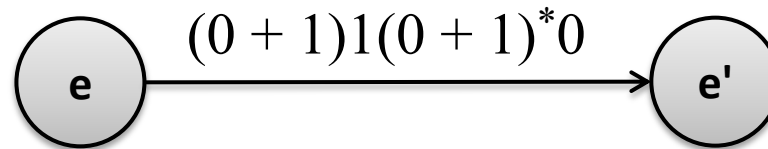
AF para $\{a\}$

b) Dados AFs para L_1 e L_2 é possível construir AFs para $L_1 \cup L_2$, $L_1 L_2$, L_1^*



Diagrama ER

- Um **diagrama ER** sobre Σ é um diagrama de estados cujas arestas, ao invés de serem rotuladas com símbolos do alfabeto Σ , são rotuladas com ERs sobre Σ



- Do estado e há transição para e' sob w se e somente se $w \in L: ((0 + 1)1(0 + 1)^*0)$, ou seja, o segundo símbolo de w é 1 e o último é 0
 - Ex.: As menores palavras são 010 e 110



Expressões Regulares

- Toda **linguagem regular** é denotada por alguma **expressão regular**

Seja um AFD $M = (E, \Sigma, \delta, i, \{f_1, f_2, \dots, f_n\})$

- $L_1(M) \cup L_2(M) \cup \dots \cup L_n(M)$, tal que

$$L_k(M) = \{w \in \Sigma^* \mid \delta(i, w) = f_k\}$$

- Seja p_k uma expressão regular para $L_k(M)$, então $L(M)$ seria denotada por $p_1 + p_2 + \dots + p_n$



Obtenção de Expressão Regular

- Para obter uma expressão regular a partir de um AFN
 $M = (E, \Sigma, \delta, I, F)$
 1. Obter AFN $M' = (E \cup \{i', f_k\}, \Sigma, \delta', \{i'\}, \{f_k\})$ equivalente à M tal que
 - a) $i \in \delta'(i', \lambda)$ para todo $i \in I$
 - b) $i' \notin \delta'(e, a)$ para todo par $(e, a) \in E' \times \Sigma$
 - c) $f_k \in \delta'(f, \lambda)$ para todo $f \in F$
 - d) $\delta'(f_k, a) = \emptyset$ para todo $a \in \Sigma$
 2. Obter diagrama ER a partir de M' substituindo transições de e para e' sob s_1, s_2, \dots, s_n por uma só transição de e para e' sob $s_1 + s_2 + \dots + s_n$



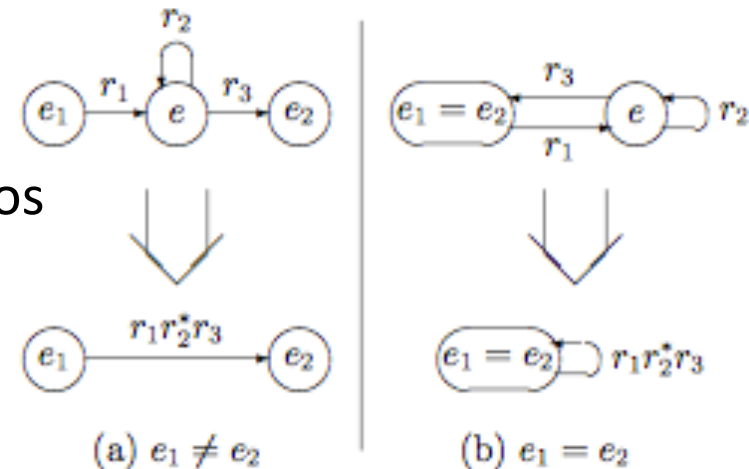
Obtenção de Expressão Regular

- Para obter uma expressão regular a partir de um AFN

$$M = (E, \Sigma, \delta, I, F)$$

3. Eliminar um a um os estados do diagrama ER com exceção dos estados i' e f_k

- Simular todas as passagens por e para cada par de estados $[e_1, e_2]$ tais que há transição de e_1 para e e de e para e_2 ($e_1 \neq e$ e $e_2 \neq e$)

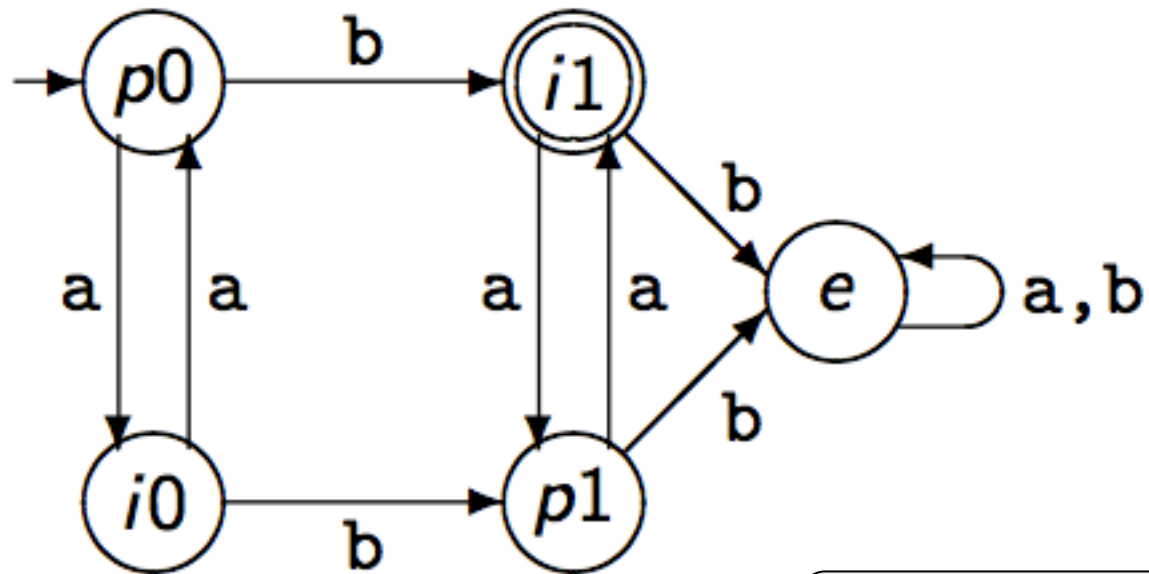


- Se havia transição de e_1 para e_2 sob s , substituir por transição de e_1 para e_2 sob $s + r_1r_2^*r_3$



Exemplo

- Considere o AFD que reconhece a linguagem
 $L = \{w \in \{a,b\}^* \mid |w| \text{ é ímpar e } w \text{ contém exatamente um } b\}$

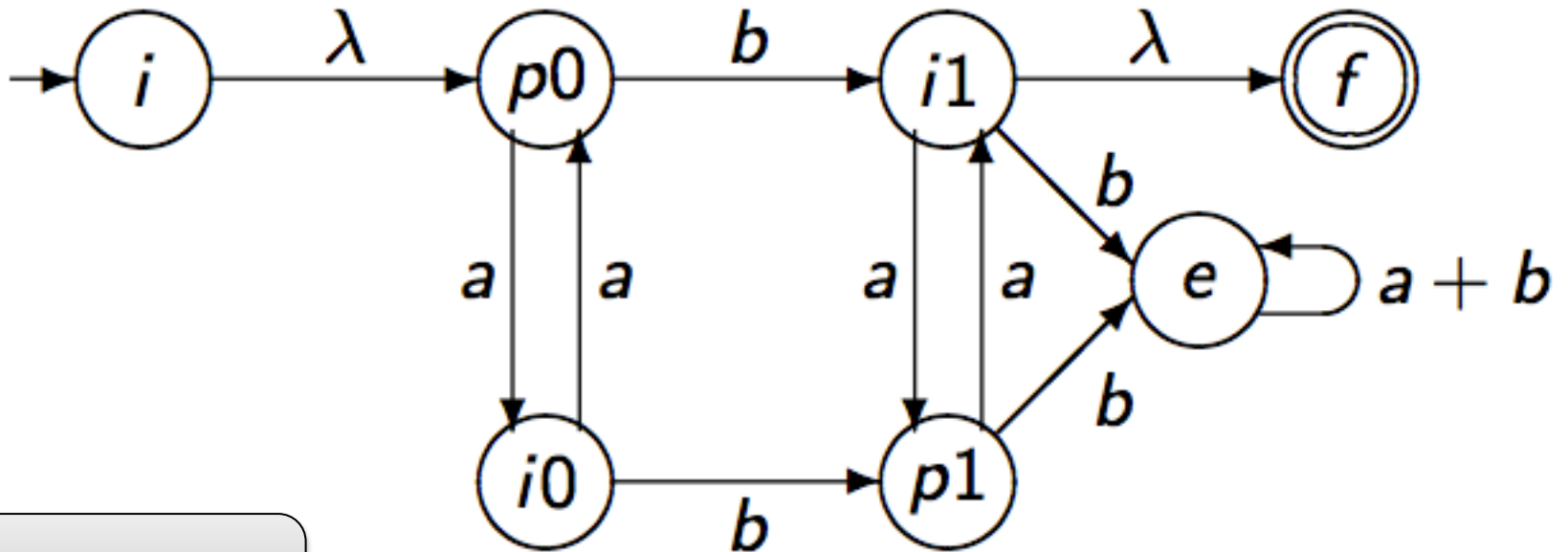


Como obter o diagrama ER com transições λ ?



Exemplo

- Considere o AFD que reconhece a linguagem
 $L = \{w \in \{a,b\}^* \mid |w| \text{ é ímpar e } w \text{ contém exatamente um } b\}$

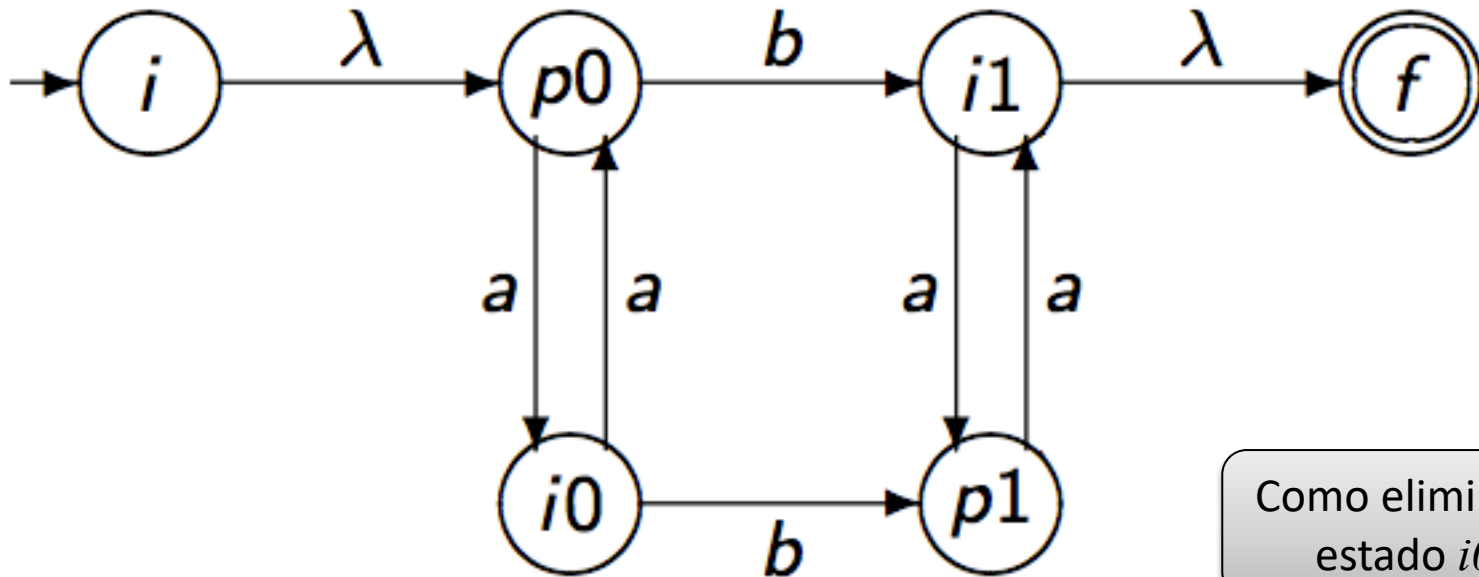


Como eliminar
o estado e ?



Exemplo

- Considere o AFD que reconhece a linguagem
 $L = \{w \in \{a,b\}^* \mid |w| \text{ é ímpar e } w \text{ contém exatamente um } b\}$

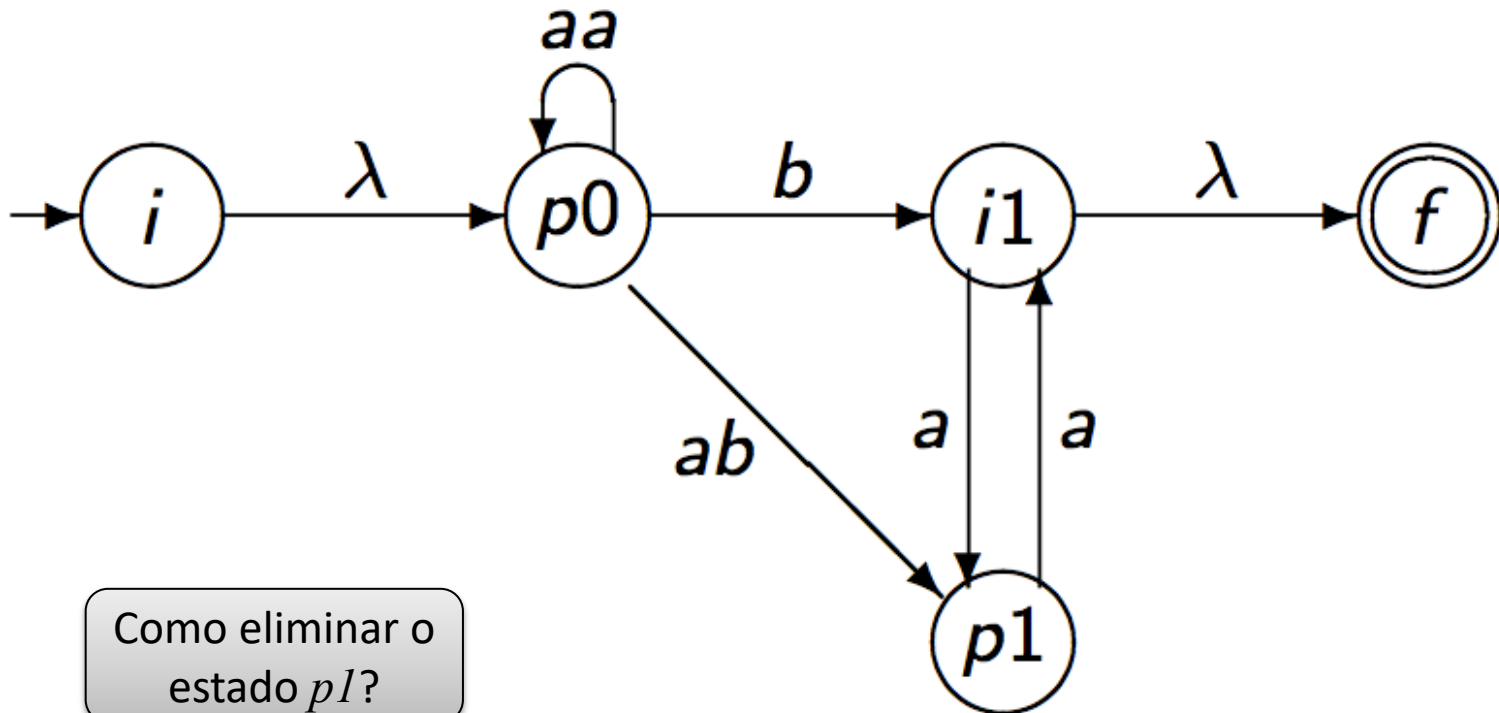


Como eliminar o estado i_0 ?



Exemplo

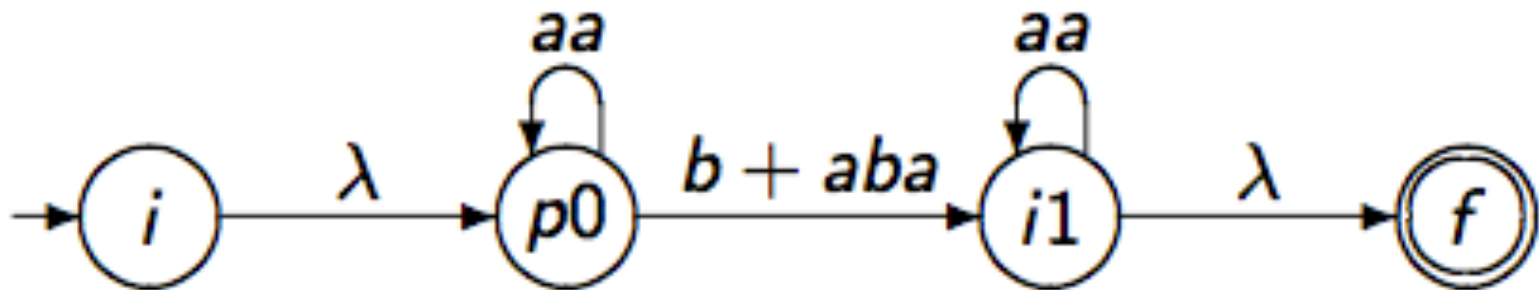
- Considere o AFD que reconhece a linguagem
 $L = \{w \in \{a,b\}^* \mid |w| \text{ é ímpar e } w \text{ contém exatamente um } b\}$





Exemplo

- Considere o AFD que reconhece a linguagem
 $L = \{w \in \{a,b\}^* \mid |w| \text{ é ímpar e } w \text{ contém exatamente um } b\}$

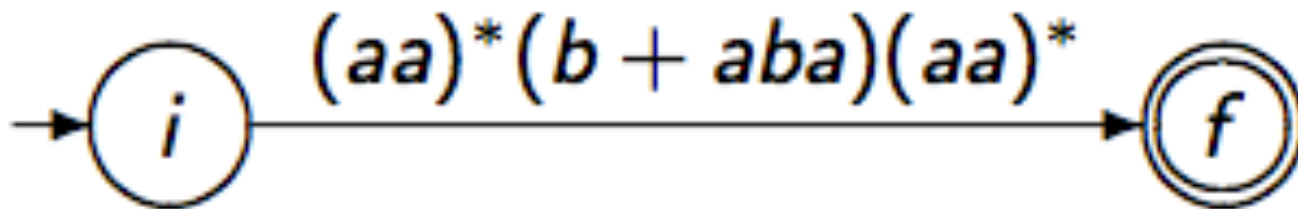


Como eliminar os estados $p0$ e $i1$?



Exemplo

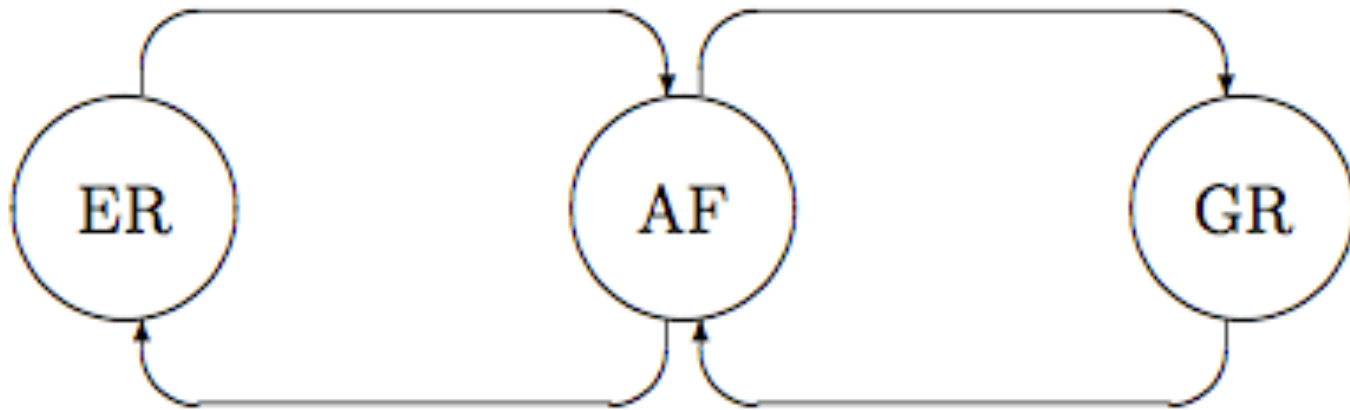
- Considere o AFD que reconhece a linguagem
 $L = \{w \in \{a,b\}^* \mid |w| \text{ é ímpar e } w \text{ contém exatamente um } b\}$





Síntese

- AFs, ERs e GRs são formalismos alternativos para **linguagens regulares**



Transformações entre formalismos



CEFET-MG

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS

ISSO É TUDO, PESSOAL!



Linguagens Formais e Autômatos